

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A computer-implemented method for bypassing I/O operations of a file system included in said computer, ~~said computer having a computer program application that includes ordered computer code, said ordered code including I/O access commands, said file system that is optimized for processing queued said I/O access commands, and said computer system having application programming interfaces and a shell interface that enable bypassing said I/O operations, the method comprising:~~

including in said computer an operating system having application programming interfaces and a shell interface;

ordering computer code that includes I/O access commands in an application;

when said file system is optimized for processing queued I/O access commands that are a type of said I/O operations;

locating asynchronous direct said I/O access commands that are included in said application ordered computer code; and

bypassing said file system queued I/O access commands when porting said application from said operating system to a different operating system ~~said~~

~~files system~~ by executing said asynchronous direct I/O access commands by

17 use of said application programming interfaces and ~~said~~ shell interface.

1 2. (Canceled)

1 3. (Currently Amended) The computer-implemented method of Claim 1, further comprising  
2 bypassing said file system queued I/O access commands by use of a performance file.

1 4. (Currently Amended) A computer-implemented method for aggregating asynchronous direct  
2 I/O access commands, ~~said computer having a computer program application that does~~  
3 ~~application I/O caching and includes ordered computer code, said each ordered computer code~~  
4 ~~having at least one said asynchronous direct I/O access command that operates with said~~  
5 ~~application I/O caching, said computer supporting I/O request chaining, said computer having a~~  
6 ~~file system that locates storage space for said computer code on said disk, said computer that~~  
7 ~~executes said computer program application, the method comprising:~~

8 including in said computer an operating system having application programming interfaces  
9 and a shell interface;

10 ordering computer code that includes I/O access commands in an application that does I/O  
11 caching;

12 supporting I/O request chaining in said computer that includes a file system;

13 when said file system is optimized for processing queued I/O access commands;

14 ~~locating at least one asynchronous direct I/O access command;~~

15 associating ~~at least one~~ asynchronous direct I/O access commands with at least one  
16 file in said file system;  
17 associating said at least one file with at least one performance file;  
18 chaining said asynchronous direct I/O access commands into at least one aggregated  
19 I/O access command in said computer program application;  
20 associating said at least one aggregated I/O access command with said at least one  
21 performance file;  
22 identifying a terminus point in said ordered computer code;  
23 issuing said at least one aggregated I/O access command by use of said application  
24 programming interfaces and shell interface until said terminus point is  
25 reached; and  
26 when said terminus point is reached and if said at least one aggregated I/O command  
27 remains, issuing a final said at least one aggregated I/O access command.

1 5. (Currently Amended) The computer-implemented method of Claim 4, further comprising:

2 including data in said ~~at least one~~ asynchronous direct I/O access commands; and  
3 including said data in said at least one aggregated I/O access command.

1 6. (Previously Presented) The computer-implemented method of Claim 4, further comprising

2 allocating said performance file in single extents.

1     7. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2     pre-formatting said performance file.

1     8. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2     allocating said performance file in a named performance file pool.

1     9. (Previously Presented) The computer-implemented method of Claim 8, further comprising  
2     marking said performance file in said performance file pool as free.

1     10. (Previously Presented) The computer-implemented method of Claim 8, further comprising  
2     marking said performance file in said performance file pool as used.

1     11. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2     allocating said performance file in a default performance file pool.

1     12. (Previously Presented) The computer-implemented method of Claim 11, further comprising  
2     marking said performance file in said default performance file pool as free.

1     13. (Previously Presented) The computer-implemented method of Claim 11, further comprising  
2     marking said performance file in said default performance file pool as used.

1 14. (Previously Presented) The computer-implemented method of Claim 4, further comprising  
2 manipulating said performance file by a file pool utility.

1 15. (Original) The computer-implemented method of Claim 4, further comprising recovering  
2 from errors occurring while executing said at least one aggregated I/O access command.

1 16. (Currently Amended) The computer-implemented method of Claim 4, further comprising  
2 locating said ~~at least one~~ asynchronous direct I/O access commands in a loop in said ordered  
3 computer code.

1 17. (Currently Amended) A computer system for bypassing I/O operations of a file system  
2 included in said computer system, ~~said computer system having a computer program application~~  
3 ~~that includes ordered computer code, said ordered code including I/O access commands, said file~~  
4 ~~system that is optimized for processing queued said I/O access commands, and said computer~~  
5 ~~system having application programming interfaces and a shell interface that enable bypassing~~  
6 ~~said I/O operations, comprising:~~

7 an operating system having application programming interfaces and a shell interface;

8 ordered computer code that includes I/O access commands in an application;

9 when said file system is optimized for processing queued said I/O access commands that are

10 a type of said I/O operations;

11 asynchronous direct ~~said~~ I/O access commands that are included in said application

12                   ordered computer code; and  
13                   said queued I/O access commands ~~file system that are~~ is bypassed when said  
14                   application is ported from said operating system to a different operating  
15                   system by executing said asynchronous direct I/O access commands by use of  
16                   said application programming interfaces and ~~said~~ shell interface.

18. (Canceled)

1                   19. (Currently Amended) The computer system of Claim 17, further comprising said queued  
2                   I/O access commands ~~file system that are~~ is bypassed by use of a performance file.

1                   20. (Currently Amended) A computer system for aggregating asynchronous direct I/O access  
2                   commands, ~~said computer having a computer program application that does application I/O~~  
3                   ~~caching and includes ordered computer code, said each ordered computer code having at least~~  
4                   ~~one said asynchronous direct I/O access command that operates with said application I/O~~  
5                   ~~caching, said computer supporting I/O request chaining, said computer having a file system that~~  
6                   ~~locates storage space for said computer code on said disk, said computer that executes said~~  
7                   ~~computer program application, comprising:~~  
8                   ~~said at least one asynchronous direct I/O access command;~~  
9                   an operating system having application programming interfaces and a shell interface;  
10                   ordered computer code that includes I/O access commands in an application that does I/O  
11                   caching;

12 a file system;

13 when said file system is optimized for processing queued I/O access commands;

14 ~~said at least one~~ asynchronous direct I/O access commands that are ~~is~~ associated with  
15 at least one file in said file system;

16 said at least one file that is associated with at least one performance file;

17 said asynchronous direct I/O access commands that ~~is~~ are chained into at least one  
18 aggregated I/O access command in said computer program application;

19 a terminus point in said ordered computer code;

20 said at least one aggregated I/O access command that is associated with said at least  
21 one performance file;

22 ~~a terminus point in said ordered computer code;~~

23 ~~said at least one aggregated I/O command~~ and that is issued until said terminus point is  
24 reached; and

25 when said terminus point is reached and if said at least one aggregated I/O command  
26 remains, a final said at least one aggregated I/O access command ~~that is issued~~.

1 21. (Currently Amended) The computer system of Claim 20, further comprising:

2 data that is included in said ~~at least one~~ asynchronous direct I/O access commands;

3 and

4 said data that is included in said at least one aggregated I/O access command.

1 22. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is allocated in single extents.

1 23. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is a pre-formatted file.

1 24. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is allocated in a named performance file pool.

1 25. (Previously Presented) The computer system of Claim 24, further comprising said  
2 performance file that is marked in said named performance file pool as free.

1 26. (Previously Presented) The computer system of Claim 24, further comprising said  
2 performance file that is marked in said named performance file pool as used.

1 27. (Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is allocated in a default performance file pool.

1 28. (Previously Presented) The computer system of Claim 27, further comprising said  
2 performance file that is marked in said default performance file pool as free.



1 29. (Previously Presented) The computer system of Claim 27, further comprising said  
2 performance file that is marked in said default performance file pool as used.

1 30.(Previously Presented) The computer system of Claim 20, further comprising said  
2 performance file that is manipulated by a file pool utility.

1 31. (Currently Amended) The computer system of Claim 20, further comprising said ~~executing~~  
2 at least one aggregated I/O access command that recovers from errors.

1 32.(Currently Amended) The computer system of Claim 20, further comprising said ~~at least one~~  
2 asynchronous direct I/O access commands that is are located in a loop in said ordered computer  
3 code.

1 33. (Canceled)

2

3 34.(Canceled)

4

5 35.(Canceled)

1 36. (Canceled)

1 37.(Canceled)

1 38. (New) An article of manufacture comprising a program storage medium readable by a  
2 computer and embodying one or more instructions executed by said computer for bypassing I/O  
3 operations of a file system included in said computer, wherein said article of manufacture is  
4 operable to:

5 include in said computer an operating system having application programming interfaces  
6 and a shell interface;

7 order computer code that includes I/O access commands in an application;

8 when said file system is optimized for processing queued I/O access commands that are a  
9 type of said I/O operations:

10 locate asynchronous direct I/O access commands in said application ordered

11 computer code; and

12 bypass said queued I/O access commands when porting said application from said

13 operating system to a different operating system by executing said

14 asynchronous direct I/O access commands by use of said application

15 programming interfaces and shell interface.

1 39. (New) The article of manufacture of Claim 38, further operable to bypass said queued I/O  
2 access commands by use of a performance file.

1     40. (New) An article of manufacture comprising a program storage medium readable by a  
2     computer and embodying one or more instructions executed by said computer for aggregating  
3     asynchronous direct I/O access commands, wherein said article of manufacture is operable to:  
4         include in said computer an operating system having application programming interfaces  
5             and a shell interface;  
6         order computer code that includes I/O access commands in an application that does I/O  
7             caching;  
8         support I/O request chaining in said computer that includes a file system;  
9         when said file system is optimized for processing queued I/O access commands:  
10             associate said asynchronous direct I/O access commands with at least one file in  
11                 said file system;  
12             associate said at least one file with at least one performance file;  
13             chain said asynchronous direct I/O access commands into at least one aggregated  
14                 I/O access command in said computer program application;  
15             associate said at least one aggregated I/O access command with said at least one  
16                 performance file;  
17             identify a terminus point in said ordered computer code;  
18             issue said at least one aggregated I/O access command by use of said application  
19                 programming interfaces and shell interface until said terminus point is  
20                 reached; and

21                   when said terminus point is reached and if said at least one aggregated I/O  
22                   command remains, issue a final said at least one aggregated I/O access  
23                   command.